Zero-Shot Video Object Segmentation via Attentive Graph Neural Networks

Wenguan Wang[1*], Xiankai Lu[1*], Jianbing Shen[1†], David Crandall[2], Ling Shao[1]
[1] Inception Institute of Artificial Intelligence, UAE [2] Indiana University, USA
{wenguanwang.ai, carrierlxk, shenjianbingcg}@gmail.com
https://github.com/carrierlxk/AGNN

Abstract

This work proposes a novel attentive graph neural network (AGNN) for zero-shot video object segmentation (ZVOS). The suggested AGNN recasts this task as a process of iterative information fusion over video graphs. Specifically, AGNN builds a fully connected graph to efficiently represent frames as nodes, and relations between arbitrary frame pairs as edges. The underlying pair-wise relations are described by a differentiable attention mechanism. Through parametric message passing, AGNN is able to efficiently capture and mine much richer and higher-order relations between video frames, thus enabling a more complete understanding of video content and more accurate foreground estimation. Experimental results on three video segmentation datasets show that AGNN sets a new state-of-the-art in each case. To further demonstrate the generalizability of our framework, we extend AGNN to an additional task: image object co-segmentation (IOCS). We perform experiments on two famous IOCS datasets and observe again the superiority of our AGNN model. The extensive experiments verify that AGNN is able to learn the underlying semantic/appearance relationships among video frames or related images, and discover the common objects.

1. Introduction

Automatically identifying the primary objects in videos is an important problem that could benefit a wide variety of applications, by reducing or eliminating manual effort needed to process and understand video. However, discovering the most prominent and distinct objects across video frames without having prior knowledge of what those foreground objects are is a challenging task. Traditional methods tend to tackle this issue by using handcrafted or learnable features in a *local* or *sequential* manner. For instance, handcrafted feature based methods use objectness [74], motion boundary [43], and saliency [67] cues over a few successive video frames, or explore trajectories [41], *i.e.*, link optical flow over multiple frames to capture long-term motion information. These are typically non-learning methods working in a purely *unsupervised* manner. Recent deep learning based methods learn more powerful video object features from large-scale training data, yielding a *zero-shot* solution [63] (still no annotation used for any testing frame). Many of these [7, 57, 21, 58, 31, 55] employ two-stream networks to combine local motion and appearance information, and apply recurrent neural networks to model the dynamics in a frame-by-frame manner.

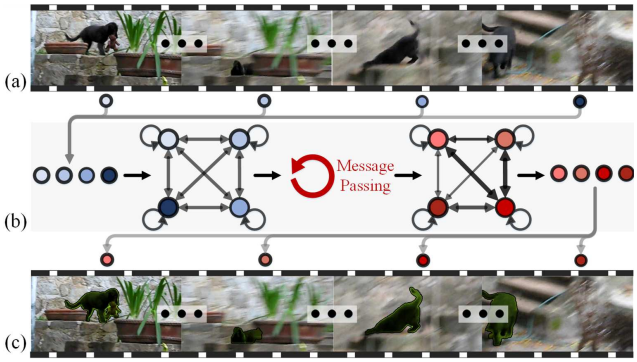

Figure 1: **Illustration of the proposed AGNN based ZVOS model.** (a) Input video sequence, typically with object occlusion and scale variation. (b) The suggested AGNN represents video frames as nodes (blue circles), and the relations between arbitrary frame pairs as edges (black arrows), captured by an attention mechanism. After several message passing iterations, higher-order relations can be mined and more optimal foreground estimations are obtained from a global view. (c) Final video object segmentation results. Best viewed in color. Zoom in for details.

Though these methods greatly promoted the development of this field and gained promising results, they generally suffer from two limitations. First, they focus primarily on the local pair-wise or sequential relations between successive frames, while ignoring the ubiquitous, high-order relationships among the frames (since frames from the same video are usually correlated). Second, since they do not fully leverage the rich relationships, they fail to completely capture the video content and hence may easily get inferior

*The first two authors contribute equally to this work.
†Corresponding author: *Jianbing Shen*.

foreground estimates. From another perspective, as video objects usually suffer from underlying object occlusions, huge scale variations and appearance changes (Fig. 1 (a)), it is difficult to correctly infer the foreground when only considering successive or local pair-wise relations in videos.

To alleviate these issues, we need to explore an effective framework that can comprehensively model the high-order relationships among video frames into modern neural networks. In this work, an attentive graph neural network (AGNN) is proposed to addresses zero-shot video object segmentation (ZVOS), which recasts ZVOS as an end-to-end, message passing based graph information fusion procedure (Fig. 1 (b)). Specifically, we construct a fully connected graph where video frames are represented as nodes and the pair-wise relations between two frames are described as the edge between their corresponding nodes. The correlation between two frames is efficiently captured by an attention mechanism, which avoids time-consuming optical flow estimation [7, 57, 21, 58, 31]. By using recursive message passing to iteratively propagate information over the graph, *i.e*., each node receives the information from other nodes, AGNN can capture higher-order relationships among video frames and obtain more optimal results from a global view. In addition, as video object segmentation is a per-pixel prediction task, AGNN has a desirable, spatial information preserving property, which significantly distinguishes it from previous fully connected graph neural networks (GNNs).

AGNN operates on multiple frames, bringing the added advantage of natural training data augmentation, as the combination candidates are numerous. In addition, since AGNN offers a powerful tool for representing and mining much richer and higher-order relationships among video frames, it brings a more complete understanding of video content. More significantly, due to its recursive property, AGNN is flexible enough to process variable numbers of nodes during inference, enabling it to consider more input information and gain better performance (Fig. 1 (c)).

We extensively evaluate AGNN on three widely-used video object segmentation datasets, namely DAVIS_{16} [45], Youtube-Objects [47] and DAVIS_{17} [46], showing its superior performance over current state-of-the-art methods.

AGNN is a fully differential, end-to-end trainable framework that allows rich and high-order relations among frames (images) to be captured and is highly applicable to spatial prediction problems. To further demonstrate its advantages and generalizability, we apply AGNN to an additional task: image object co-segmentation (IOCS), which aims to extract the common objects from a group of semantically related images. It also gains promising results on two popular IOCS benchmarks, PASCAL VOC [11] and Internet [51], compared to existing IOCS methods.

Experiments on the ZVOS and additional IOCS tasks clearly demonstrate that AGNN is able to not only capture the relationships among correlated video frame images, but also mine the semantics among semantically related static images. Notably, this work can be viewed as a very early attempt to apply and extend GNNs for pixel-wise prediction tasks, which provides an effective video object segmentation solution and new insight into this task.

2. Related Work

2.1. Graph Neural Networks

GNN was first proposed in [15] and further developed in [53] to handle the underlying relationships among structured data. In [53], recurrent neural networks were used to model the state of each node, and the underlying correlation between nodes are learned via parameterized message passing over neighbors. Li *et al*. [33] further adapted GNN to sequential outputs. Gilmer *et al*. [14] Later formulated the message passing module in GNNs as a learnable neural network. Recently, GNNs have been successfully applied in many fields, including molecular biology [14], computer vision [48, 71, 76], machine learning [62] and natural language processing [2]. Another popular trend in GNNs is to generalize the convolutional architecture over arbitrary graph-structured data [10, 40, 26], which is called graph convolution neural network (GCNN).

The proposed AGNN falls into the former category; it is a message passing based GNN, where all the nodes, edges, and message passing functions are parameterized by neural networks. It shares the general idea of mining relationships over graphs but has significant differences. First, our AGNN is unique in its spatial information preserving nature, which is opposed to conventional fully connected GNNs and crucial for per-pixel prediction task. Second, to efficiently capture the relationship between two image frames, we introduce a differentiable attention mechanism which addresses the correlated information and produces further discriminative edge features. Third, as far as we know, there is no prior attempt to explore GNNs in ZVOS.

2.2. Automatic Video Object Segmentation

To automatically separate primary objects from the background, *conventional* methods typically use handcrafted features (*e.g*., color, optical flow) [43, 12, 59, 20] and certain heuristic assumptions related to the foreground (*i.e*., local motion differences [43], background priors [67]). Some others explore more efficient object representations, such as dense point trajectories [41, 42, 68] or object proposals [74, 27, 23, 36]. Most of these methods work in a purely unsupervised manner without using any training data.

Recently, with the renaissance of deep learning, more research efforts have been devoted to tackling this in deep learning frameworks, leading to a zero-shot solution [13, 21, 58, 7, 30, 31, 29, 37]. For instance, a multi-layer perception based detector was designed in [13] to detect moving

objectness. Li *et al*. [30] integrated deep learning based instance embedding and motion saliency [30] to boost performance. Some others turned to fully convolutional networks (FCNs) [3, 34, 77]. They introduced two-stream networks to fuse appearance and motion information [29, 21, 7], or explored more efficient feature extraction models and LSTM variants [55], to better locate the foreground objects.

The differences from previous methods are multifold: our AGNN **1)** provides a unified, end-to-end trainable, graph model based ZVOS solution; **2)** efficiently mines diverse and high-order relations within videos, through iteratively propagating and fusing messages over the graph; and **3)** utilizes a differentiable attention mechanism to capture the correlated information between frame pairs.

2.3. Image Object Co-Segmentation

IOCS [50, 39, 18] aims to jointly segment common objects belonging to the same semantic class in a given set of related images. Early methods usually formulate IOCS as an energy function defined over the whole or a part of the image set and consider intra- and inter-image cues [64, 25, 52, 65]. To capture the relationships between images, some methods applied scene matching techniques [51], global appearance models [66], discriminative clustering methodologies [22], manifold ranking [49] or saliency heuristics [16, 56]. There are only a very few deep IOCS models [4, 32], mainly due to the lack of a proper, end-to-end modeling strategy for this problem. [4, 32] tackled IOCS through a pair-wise comparison protocol and employed a Siamese network to capture the similarity between two related images. Our AGNN based ICOS solution is significantly different from [4, 32]. First, [4, 32] consider IOCS as a pair-wise image matching problem, while we formulate IOCS as an information propagation and fusion process among multiple images. That means our model can capture richer relations from a global view. Second, the Siamese network based systems only handle pair-wise relations, while our message passing based iterative inference can learn higher-order relations among multiple images. Third, our method is based on the graph model, yielding a more general and elegant framework for modeling IOCS.

3. Our Algorithm

Before elaborating on our proposed AGNN (§3.2), we first give a brief introduction to generic formulations of GNN models (§3.1). Finally, in §3.3, we provide detailed information on our network architecture.

3.1. General Formulations of GNNs

Based on deep neural networks and graph theory, GNNs are powerful for collectively aggregating information from data represented in graph domains [53, 14]. Specifically, a GNN model is defined according to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node $v_i \in \mathcal{V}$ takes a unique value from $\{1, \ldots, |\mathcal{V}|\}$, is associated with an initial *node representation* (or *node state* or *node embedding*) $\mathbf{v}_i$. Each edge $e_{i,j} \in \mathcal{E}$ is a pair $e_{i,j} = (v_i, v_j) \in |\mathcal{V}| \times |\mathcal{V}|$, with an *edge representation* $\mathbf{e}_{i,j}$. For each node v_i, we learn an updated node representation $\mathbf{h}_i$ through aggregating representations of its neighbors. Here $\mathbf{h}_i$ is used to produce an output $\mathbf{o}_i$, *i.e*., a node label. More specifically, GNNs map graph $\mathcal{G}$ to the node outputs $\{\mathbf{o}_i\}_{i=1}^{|\mathcal{V}|}$ through two phases. First, a parametric *message passing phase* runs for K steps, which recursively propagates messages and updates node representations. At the k-th iteration, for each node v_i, we update its state according to its received message $\mathbf{m}_i^k$ (*i.e*., summarized information from its neighbors $\mathcal{N}_i$) and its previous state $\mathbf{h}_i^{k-1}$:

$$\begin{aligned} \text{message aggregation:}\quad \mathbf{m}_i^k &= \sum\nolimits_{v_j \in \mathcal{N}_i} \mathbf{m}_{j,i}^k, \\ &= \sum\nolimits_{v_j \in \mathcal{N}_i} M(\mathbf{h}_j^{k-1}, \mathbf{e}_{i,j}^{k-1}), \\ \text{node representation update:}\quad \mathbf{h}_i^k &= U(\mathbf{h}_i^{k-1}, \mathbf{m}_i^k), \end{aligned} \tag{1}$$

where $\mathbf{h}_i^0 = \mathbf{v}_i$, $M(\cdot)$ and $U(\cdot)$ are the *message function* and *state update function*, respectively. After k iterations of aggregation, $\mathbf{h}_i^k$ captures the relations within the k-hop neighborhood of node v_i.

Second, a *readout phase* maps the node representation $\mathbf{h}_i^K$ of the final K-iteration to a node output, through a *readout function* $R(\cdot)$:

$$\text{readout:}\quad \mathbf{o}_i = R(\mathbf{h}_i^K). \tag{2}$$

The message function M, update function U, and readout function R are all learned differentiable functions.

Next, we present our AGNN based ZVOS solution, which essentially extends traditional fully connected GNNs to (1) preserve spatial features; and (2) capture pair-wise relations (edges) via a differentiable attention mechanism.

3.2. Attentive Graph Neural Network

Problem Definition and Notations. Given a set of training samples and an unseen testing video $\mathcal{I} = \{I_i \in \mathbb{R}^{w \times h \times 3}\}_{i=1}^N$ with N frames in total, the goal of ZVOS is to generate a corresponding sequence of binary segment masks: $\mathcal{S} = \{S_i \in \{0,1\}^{w \times h}\}_{i=1}^N$. To achieve this, AGNN represents $\mathcal{I}$ as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where node $v_i \in \mathcal{V}$ represents the i-th frame I_i, and edge $e_{i,j} = (v_i, v_j) \in \mathcal{E}$ indicates the relation from I_i to I_j. To comprehensively capture the underlying relationships between video frames, we assume $\mathcal{G}$ is fully connected and includes self-connections at each node (see Fig. 2 (a)). For clarity, we refer to $e_{i,i}$, which connects a node v_i to itself, as a *loop-edge*; and $e_{i,j}$, which connects two different nodes v_i and v_j, as a *line-edge*.

The core idea of our AGNN is to perform K message propagation iterations over $\mathcal{G}$ to efficiently mine rich and high-order relations within $\mathcal{I}$. This helps to better capture

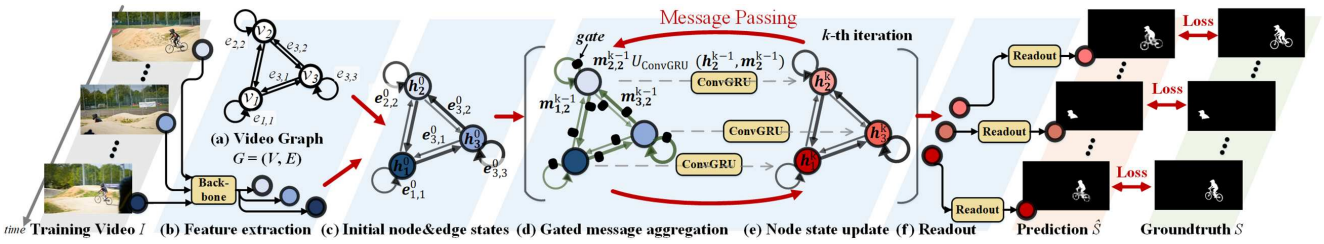

Figure 2: Our AGNN based ZVOS model during the training phase (see §3.2 and §3.3). Zoom in for details.

the video content from a global view and obtain more accurate foreground estimates. We then readout the segmentation predictions $\hat{\mathcal{S}}$ from the final node states $\{\mathbf{h}_i^K\}_{i=1}^N$. Next, we describe each component of our model in detail.

FCN-Based Node Embedding. We leverage DeepLabV3 [5], a classical FCN based semantic segmentation architecture, to extract effective frame features, as node representations (see Fig. 2 (b) and Fig. 3 (a)). For node v_i, its initial embedding $\mathbf{h}_i^0$ can be computed as:

$$\mathbf{h}_i^0 = \mathbf{v}_i = F_{\text{DeepLab}}(I_i) \in \mathbb{R}^{W\times H\times C}, \tag{3}$$

where $\mathbf{h}_i^0$ is a 3D tensor feature with $W\times H$ spatial resolution and C channels, which preserves spatial information as well as high-level semantic information.

Intra-Attention Based Loop-Edge Embedding. A loop-edge $e_{i,i}\in\mathcal{E}$ is a special edge that connects a node to itself. The loop-edge embedding $\mathbf{e}_{i,i}^k$ is used to capture the intra relations within node representation $\mathbf{h}_i^k$ (*i.e.*, internal frame representation). We formulate $\mathbf{e}_{i,i}^k$ as an *intra-attention* mechanism [61, 70], which has been proven complementary to convolutions and helpful for modeling long-range, multi-level dependencies across image regions [75]. In particular, the intra-attention calculates the response at a position by attending to all the positions within the same node embedding (see Fig. 2 (c) and Fig. 3 (b)):

$$\begin{aligned}\mathbf{e}_{i,i}^k &= F_{\text{intra-att}}(\mathbf{h}_i^k) \in \mathbb{R}^{W\times H\times C}\\ &= \alpha\ \text{softmax}\big((\mathbf{W}_f * \mathbf{h}_i^k)(\mathbf{W}_h * \mathbf{h}_i^k)^\top\big)(\mathbf{W}_l * \mathbf{h}_i^k) + \mathbf{h}_i^k,\end{aligned} \tag{4}$$

where '$*$' represents the convolution operation, $\mathbf{W}$s indicate learnable convolution kernels, and α is a learnable scale parameter. Eq. 4 makes the output element of each position in $\mathbf{h}_i^k$ encode contextual information as well as its original information, thus enhancing the representability.

Inter-Attention Based Line-Edge Embedding. A line-edge $e_{ij}\in\mathcal{E}$ connects two different nodes v_i and v_j. The line-edge embedding $\mathbf{e}_{i,j}^k$ is used to mine the relation from node v_i to v_j, in the node embedding space (see Fig. 2 (b)). Here we compute an *inter-attention* mechanism [35] to capture the bi-directional relations between two nodes v_i and v_j (see Fig. 2 (c) and Fig. 3 (c)):

$$\begin{aligned}\mathbf{e}_{i,j}^k &= F_{\text{inter-att}}(\mathbf{h}_i^k, \mathbf{h}_j^k) = \mathbf{h}_i^k \mathbf{W}_c \mathbf{h}_j^{k\top} \in \mathbb{R}^{(WH)\times(WH)},\\ \mathbf{e}_{j,i}^k &= F_{\text{inter-att}}(\mathbf{h}_j^k, \mathbf{h}_i^k) = \mathbf{h}_j^k \mathbf{W}_c^\top \mathbf{h}_i^{k\top} \in \mathbb{R}^{(WH)\times(WH)},\end{aligned} \tag{5}$$

where $\mathbf{e}_{i,j}^k = \mathbf{e}_{j,i}^{k\top}$. $\mathbf{e}_{i,j}^k$ indicates the outgoing edge feature and $\mathbf{e}_{j,i}^k$ the incoming one, for node v_i. $\mathbf{W}_c\in\mathbb{R}^{C\times C}$ indicates a learnable weight matrix. $\mathbf{h}_j^k\in\mathbb{R}^{(WH)\times C}$ and $\mathbf{h}_i^k\in\mathbb{R}^{(WH)\times C}$ are flattened into matrix representations. Each element in $\mathbf{e}_{i,j}^k$ reflects the similarity between each row of $\mathbf{h}_i^k$ and each column of $\mathbf{h}_j^{k\top}$. As a result, $\mathbf{e}_{i,j}^k$ can be viewed as the *importance* of node v_i's embedding to v_j, and vice versa. By attending to each node pair, $\mathbf{e}_{i,j}^k$ explores their joint representations in the node embedding space.

Gated Message Aggregation. In our AGNN, for the message passed in the self-loop, we view the loop-edge embedding $\mathbf{e}_{i,j}^{k-1}$ itself as a message (see Fig. 3 (b)), since it already contains the contextual and original node information (see Eq. 4):

$$\mathbf{m}_{i,i}^k = \mathbf{e}_{i,i}^{k-1} \in \mathbb{R}^{W\times H\times C}. \tag{6}$$

For the message $\mathbf{m}_{j,i}$ passed from node v_j to v_i (see Fig. 3 (c)), we have:

$$\mathbf{m}_{j,i}^k = M(\mathbf{h}_j^{k-1}, \mathbf{e}_{i,j}^{k-1}) = \text{softmax}(\mathbf{e}_{i,j}^{k-1})\mathbf{h}_j^{k-1} \in \mathbb{R}^{(WH)\times C}, \tag{7}$$

where softmax$(\cdot)$ normalizes each row of the input. Thus, each row (position) of $\mathbf{m}_{j,i}^k$ is a weighted combination of each row (position) of $\mathbf{h}_j^{k-1}$, where the weights come from the corresponding column of $\mathbf{e}_{i,j}^{k-1}$. In this way, the message function $M(\cdot)$ assigns its edge-weighted feature (*i.e.*, message) to the neighbor nodes [62]. Then, $\mathbf{m}_{j,i}^k$ is reshaped back to a 3D tensor with a size of $W\times H\times C$.

In addition, because some nodes are noisy due to camera shift or out-of-view, their messages may be useless or even harmful. We apply a learnable gate $G(\cdot)$ to measure the confidence of a message $\mathbf{m}_{j,i}$:

$$\mathbf{g}_{j,i}^k = G(\mathbf{m}_{j,i}^k) = \sigma\big(F_{\text{GAP}}(\mathbf{W}_g * \mathbf{m}_{j,i}^k + b_g)\big) \in [0,1]^C, \tag{8}$$

where $F_{\text{GAP}}(\cdot)$ indicates the use of global average pooling to generate channel-wise responses, σ is the logistic sigmoid function $\sigma(x) = 1/(1+\exp(-x))$, and $\mathbf{W}_g$ and b_g are the trainable convolution kernel and bias.

Following Eq. 1, we collect the messages from the neighbors and self-loop via gated summarization (see Fig. 2 (d)):

$$\mathbf{m}_i^k = \sum\nolimits_{v_j\in\mathcal{V}} \mathbf{g}_{j,i}^k \star \mathbf{m}_{j,i}^k \in \mathbb{R}^{W\times H\times C}, \tag{9}$$

where '$\star$' denotes the channel-wise Hadamard product. Here, the gate mechanism is used to filter out irrelevant information from noisy frames. See §4.3 for a quantitative study of this design.

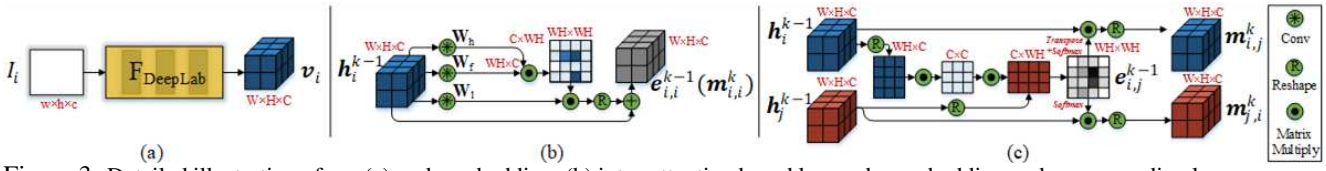

Figure 3: Detailed illustration of our (a) node embedding, (b) intra-attention based loop-edge embedding and corresponding loop-message generation, (c) inter-attention based straight-edge embedding and corresponding neighbor message generation.

ConvGRU based Node-State Update. In step k, after aggregating all the information from the neighbor nodes and itself (Eq. 9), v_i gets a new state $\mathbf{h}_i^k$ by taking into account its prior state $\mathbf{h}_i^{k-1}$ and its received message $\mathbf{m}_i^k$. To preserve the spatial information conveyed in $\mathbf{h}_i^{k-1}$ and $\mathbf{m}_i^k$, we leverage ConvGRU [1] to update the node state (Fig. 2 (e)):

$$\mathbf{h}_i^k = U_{\text{ConvGRU}}(\mathbf{h}_i^{k-1}, \mathbf{m}_i^k) \in \mathbb{R}^{W\times H\times C}. \quad (10)$$

ConvGRU is proposed as a convolutional counterpart to previous fully connected GRU [9], and introduces convolution operation into input-to-state and state-to-state transitions.

Readout Function. After K message passing iterations, we obtain the final state $\mathbf{h}_i^K$ for each node v_i. Finally, in the readout phase, we get a segmentation prediction map $\hat{S} \in [0,1]^{W\times H}$ from $\mathbf{h}_i^K$ through a readout function $R(\cdot)$ (see Fig. 2 (f)). Slightly different from Eq. 2, we concatenate the final node state $\mathbf{h}_i^K$ and the original node feature $\mathbf{v}_i$ (*i.e.*, $\mathbf{h}_i^0$) together and feed the combined feature into $R(\cdot)$:

$$\hat{S}_i = R_{\text{FCN}}([\mathbf{h}_i^K, \mathbf{v}_i]) \in [0,1]^{W\times H}. \quad (11)$$

Again, to preserve spatial information, the readout function is implemented as a small FCN network, which has three convolution layers with a sigmoid function to normalize the prediction to $[0,1]$.

The convolution operations in the intra-attention (Eq. 4) and update function (Eq. 10) are realized with 1×1 convolutional layers. The readout function (Eq. 11) consists of two 3×3 convolutional layers cascaded by a 1×1 convolutional layer. As a message passing based GNN model, these functions share weights among all the nodes. Moreover, all the above functions are carefully designed to avoid disturbing spatial information, which is essential for ZVOS since it is a pixel-wise prediction task.

3.3. Detailed Network Architecture

Our whole model is end-to-end trainable, as all the functions in AGNN are parameterized by neural networks. We use the first five convolution blocks of DeepLabV3 [5] as our backbone for feature extraction. For an input video $\mathcal{I}$, each frame I_i (with a resolution of 473×473) is represented as a node v_i in the video graph $\mathcal{G}$ and associated with an initial node state $\mathbf{v}_i = \mathbf{h}_i^0 \in \mathbb{R}^{60\times60\times256}$. Then, after a total of K message passing iterations, for each node v_i, we use the readout function in Eq. 11 to obtain a corresponding segmentation prediction map $\hat{S} \in [0,1]^{60\times60}$. More details on the training and testing phases are provided as follows.

Training Phase. As we operate on batches of a certain size (which is allowed to vary, depending on the GPU memory size), we leverage a random sampling strategy to train AGNN. Specifically, we split each training video $\mathcal{I}$ with a total of N frames into N' segments ($N' \leq N$) and randomly select one frame from each segment. Then we feed the N' sampled frames into a batch and train AGNN. Thus the relationships among all the N' sampling frames in each batch are represented using an N'-node graph. Such a sampling strategy provides robustness to variations and enables the network to fully exploit all frames. The diversity among the samples enables our model to better capture the underlying relationships and improve its generalizability. Let us denote the ground-truth segmentation mask and predicted foreground map for a training frame I_i as $S \in \{0,1\}^{60\times60}$ and $\hat{S} \in [0,1]^{60\times60}$. Our model is trained through the weighted binary cross entropy loss (see Fig. 2):

$$\mathcal{L}(S,\hat{S}) = -\sum_x^{W\times H}(1-\eta)S_x\log(\hat{S}_x) + \eta(1-S_x)\log(1-\hat{S}_x), \quad (12)$$

where η indicates the foreground-background pixel number ratio in S. It is worth mentioning that, as AGNN handles multiple video frames at the same time, it leads to a remarkably efficient training data augmentation strategy, as the combination candidates are numerous. In our experiments, during training, we randomly select 2 videos from the training video set and sample 3 frames ($N' = 3$) per video, due to the computation limitation. In addition, we set the total number of iterations as $K = 3$. Quantitative experimental settings can be found in §4.3.

Testing Phase. After training, we can apply the learned AGNN model to perform per-pixel object prediction over unseen videos. For an input test video $\mathcal{I}$ with N frames (with 473×473 resolution), we split $\mathcal{I}$ into T subsets: $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_T\}$, where $T = N/N'$. Each subset contains N' frames with an interval of T frames: $\mathcal{I}_t = \{I_t, I_{t+T}, \ldots, I_{N-T+t}\}$. Then we feed each subset into AGNN to obtain the segmentation maps of all the frames in the subset. In practice, we set $N' = 5$ during testing. We quantitatively study this setting in §4.3. As our AGNN does not require time-consuming optical flow computation and processes N' frames in one feed-forward propagation, it achieves a fast speed of $0.28s$ per frame. Following the widely used protocol [58, 57, 55], we apply CRF as a post-processing step, which takes about $0.50s$ per frame. More implementation details can be found in §4.1.1.

Method	KEY [28]	MSG [41]	NLC [12]	CUT [24]	FST [43]	SFL [7]	MP [57]	FSEG [21]	LVO [58]	ARP [27]	PDB [55]	MOA [54]	AGS [69]	**AGNN**
Mean ↑	49.8	53.3	55.1	55.2	55.8	67.4	70.0	70.7	75.9	76.2	77.2	77.2	79.7	**80.7**
$\mathcal{J}$ Recall ↑	59.1	61.6	55.8	57.5	64.9	81.4	85.0	83.0	89.1	91.1	90.1	87.8	91.1	**94.0**
Decay ↓	14.1	2.4	12.6	2.2	**0.0**	6.2	1.3	1.5	**0.0**	7.0	0.9	5.0	1.9	0.03
Mean ↑	42.7	50.8	52.3	55.2	51.1	66.7	65.9	65.3	72.1	70.6	74.5	77.4	77.4	**79.1**
$\mathcal{F}$ Recall ↑	37.5	60.0	61.0	51.9	51.6	77.1	79.2	73.8	83.4	83.5	84.4	84.4	85.8	**90.5**
Decay ↓	10.6	5.1	11.4	3.4	2.9	5.1	2.5	1.8	1.3	7.9	**-0.2**	3.3	1.6	0.03
$\mathcal{T}$ Mean ↓	26.9	30.2	42.5	27.7	36.6	28.2	57.2	32.8	**26.5**	39.3	29.1	27.9	26.7	33.7

Table 1: Quantitative results on the validation set of DAVIS$_{16}$ [45] (§4.1.2). The scores are borrowed from the public leaderboard[1]. (The best scores are marked in **bold**. The best two entries in each row are marked in gray. These notes are the same to other tables.)

	Airplane (6)	Bird (6)	Boat (15)	Car (7)	Cat (16)	Cow (20)	Dog (27)	Horse (14)	Motorbike (10)	Train (5)	Avg.
FST [43]	70.9	70.6	42.5	65.2	52.1	44.5	65.3	53.5	44.2	29.6	53.8
COSEG [60]	69.3	76.0	53.5	70.4	66.8	49.0	47.5	55.7	39.5	53.4	58.1
ARP [27]	73.6	56.1	57.8	33.9	30.5	41.8	36.8	44.3	48.9	39.2	46.2
LVO [58]	86.2	81.0	68.5	69.3	58.8	68.5	61.7	53.9	60.8	66.3	67.5
PDB [55]	78.0	80.0	58.9	76.5	63.0	64.1	70.1	67.6	58.3	35.2	65.4
FSEG [21]	81.7	63.8	72.3	74.9	68.4	68.0	69.4	60.4	62.7	62.2	68.4
SFL [7]	65.6	65.4	59.9	64.0	58.9	51.1	54.1	64.8	52.6	34.0	57.0
AGS [69]	87.7	76.7	72.2	78.6	69.2	64.6	73.3	64.4	62.1	48.2	69.7
AGNN	81.1	75.9	70.7	78.1	67.9	69.7	77.4	67.3	68.3	47.8	**70.8**

Table 2: Quantitative performance of each category on Youtube-Objects [47] (§4.1.2) with mean $\mathcal{J}$. We show the average performance for each of the 10 categories, and the final row shows an average over all the videos.

4. Experiments

We first report performance on the main task: unsupervised video object segmentation (§4.1). Then, in §4.2, to further demonstrate the advantages of our AGNN model, we test it on an additional task: image object co-segmentation. Finally, we conduct an ablation study in §4.3.

4.1. Main Task: ZVOS

4.1.1 Experimental Setup

Datasets and Metrics: We use two well-known datasets:

- **DAVIS$_{16}$ [45]** is a challenging video object segmentation dataset which consists of 50 videos in total (30 for training and 20 for val) with pixel-wise annotations for every frame. Three evaluation criteria are used in this dataset, *i.e.*, region similarity (Intersection-over-Union) $\mathcal{J}$, boundary accuracy $\mathcal{F}$, and time stability $\mathcal{T}$.
- **Youtube-Objects [47]** comprises 126 video sequences which belong to 10 object categories and contain more than 20,000 frames in total. Following its protocol, we use $\mathcal{J}$ to measure the segmentation performance.
- **DAVIS$_{17}$ [46]** consists of 60 videos in the training set, 30 videos in the validation set and 30 videos in the test-dev set. Different from DAVIS2016 and Youtube-Objects, which only focus on object-level video object segmentation, DAVIS$_{17}$ provides instance-level annotations.

Implementation Details: Following [44, 55], both static data from image salient object segmentation datasets, MSRA10K [8], DUT [72], and video data from the training set of DAVIS$_{16}$ are iteratively used to train our model. In a 'static-image' iteration, we randomly sample 6 images from the static training data to train our backbone network (DeepLabV3) to extract more discriminative foreground features. To train the backbone network, a 1×1 convolution layer with *sigmoid* function is appended as an intermediate output layer, which can access the static image supervision signal. This is followed by a 'dynamic-video' iteration, in which we use the sampling strategy described in §3.3 to sample 6 video frames to train our whole AGNN model. The 'static-image' and 'dynamic-video' iterations are executed alternately. To apply the trained AGNN model on DAVIS$_{17}$, we first use category agnostic mask-RCNN [17] to generate instance-level object proposals for each frame. Then, we run AGNN on the whole video and generate a coarse mask for the primary objects in each frame. Then the object-level masks are used to filter out the proposals from the background and highlight the foreground proposals. Through combining an instance bounding proposals and coarse masks, we obtain the instance-level mask for each primary object. Finally, to connect multiple instances across different frames, we use overlap ratio and optical flow as an association metric [38] to match different instance-level masks.

4.1.2 Quantitative Performance

Val-set of DAVIS$_{16}$. We compare the proposed AGNN with the top ZVOS methods from the DAVIS$_{16}$ benchmark[1] [45]. Table 1 shows the detailed results. We can see that our AGNN outperforms the best reported results (*i.e.*, AGS [69]) on DAVIS$_{16}$ benchmark by a significant margin in terms of mean $\mathcal{J}$ (80.7 *vs* 79.7) and $\mathcal{F}$ (79.1 *vs* 77.4). Compared to PDB [55], which uses the same training protocol and training datasets, our AGNN yields significant performance gains of 3.5% and 4.6% in terms of mean $\mathcal{J}$ and mean $\mathcal{F}$, respectively.

Youtube-Objects. Table 2 gives the detailed per-category performance and average results on Youtube-Objects. As

[1] https://davischallenge.org/davis2016/soa_compare.html, deadline: Mar. 2019

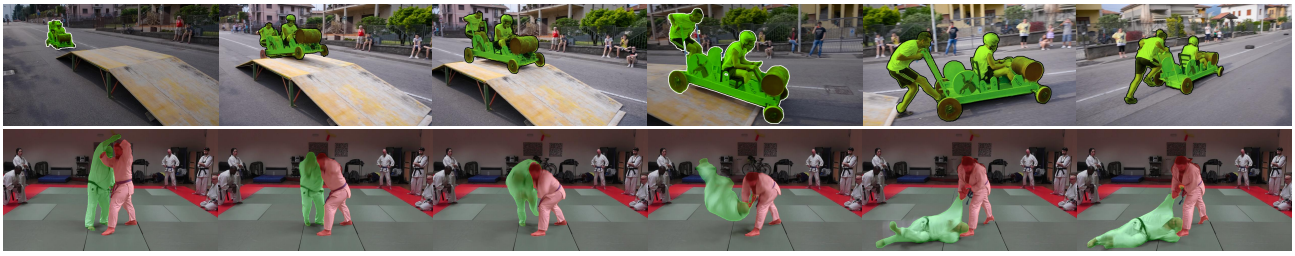

Figure 4: Qualitative results on two example videos (top: *soapbox*, bottom: *judo*) from the DAVIS_{16} val set and DAVIS_{17} test-dev set, respectively (see §4.1.3).

Method	$\mathcal{J}$			$\mathcal{F}$			$\mathcal{J}\&\mathcal{F}$
	Mean ↑	Recall ↑	Decay ↓	Mean ↑	Recall ↑	Decay ↓	Mean ↑
RVOS [63]	39.0	42.8	**0.50**	48.3	49.6	**-0.01**	43.7
AGNN	58.9	**65.7**	11.7	**63.2**	**67.1**	14.3	**61.1**

Table 3: Quantitative results on the DAVIS_{17} test-dev set [46].

can be seen, our AGNN performs favorably according to mean $\mathcal{J}$ criterion. Furthermore, unlike other methods whose performance fluctuates across categories, AGNN mains a stable performance.This further proves its robustness and generalizability.

Test-dev set of DAVIS_{17}. In Table 3 we report the performance comparison with the recent instance-level ZVOS method, RVOS [63], on the DAVIS_{17} test-dev set. We can find that AGNN significantly outperforms RVOS over most evaluation criteria.

4.1.3 Qualitative Performance

Fig. 4 depicts visual results for the proposed AGNN on two challenging video sequences *soapbox* and *judo* of DAVIS_{16} and DAVIS_{17}, respectively. For *soapbox*, the primary objects undergo huge scale variation, deformation and view changes, but our AGNN still generates accurate foreground segments. Our AGNN also handles *judo* well, although the different foreground instances suffer from similar appearance and rapid motions.

4.2. Additional Task: IOCS

Our AGNN model can be viewed as a framework for capturing high-order relations among images (or frames). To demonstrate its generalizability, we extend AGNN for IOCS task. Rather than extracting the foreground objects across multiple relatively similar video frames in videos, IOCS needs to infer the common objects from a group of semantically related images.

4.2.1 Experimental Setup

Datasets and Metrics: We perform experiments on two well-known IOCS datasets:

- **PASCAL VOC [11]** has 1,464 training images and 1,449 validation images. Following [32], we split the validation set into 724 validation and 725 test images, and use mean $\mathcal{J}$ as the performance measure.
- **Internet [51]** contains 1,306 car, 879 horse, and 561 airplane images. Following [4, 49], we measure the performance on a subset of Internet (100 images per class are sampled) with mean $\mathcal{J}$.

Method	GO-FMR [49]	FCNs [34]	CA [4]	**AGNN**
Mean $\mathcal{J}$ ↑	52.0	55.21	59.24	**60.78**
Method	FCA [4]	CSA [4]	DOCS [32]	**AGNN**
Mean $\mathcal{J}$ ↑	59.41	59.76	57.82	**60.78**

Table 4: Quantitative performance on PASCAL VOC [11] with mean $\mathcal{J}$. We show the average performance for 20 categories averaged over all the images. See §4.2.2 for detailed analyses.

Implementation Details: Following [4, 32], we employ PASCAL VOC to train our model. In each iteration, we randomly sample a group of $N'=3$ images that belong to the same semantic class, and feed two groups with randomly selected classes (6 images in total) to the network. All other experimental settings are the same as ZVOS.

After training, we evaluate the performance of our method on the test sets of PASCAL VOC and Internet dataset. When processing an image, IOCS must leverage information from the whole image group (as the images are typically different and some are irrelevant) [49, 65]. To this end, for each image I_i to be segmented, we uniformly split the other $N-1$ images into T groups, where $T=(N-1)/(N'-1)$. Then we feed the first image group and I_i to a batch of size N', and store the node state for I_i. After that, we feed the next group and the store node state of I_i to get a new state of I_i. After T steps, the final state of I_i contains its relationships to all other images and is used to produce its final co-segmentation result.

4.2.2 Quantitative Performance

PASCAL VOC. It is very challenging to segment the common objects in this dataset, since the objects undergo drastic variation in scale, position and appearance. In addition, some images have multiple objects belonging to different categories. On this dataset, we compare AGNN with six representative methods, including Siamese-based co-segmentation methods [4, 32], as well as deep semantic segmentation models (*e.g.*, FCNs [34]).

Table 4 shows detailed results in terms of mean $\mathcal{J}$. FCNs [34] segment each image individually (without considering other related images), and thus give poor performance. Both [4] and [32] consider pairs of images and gain better results. Our AGNN achieves the best performance because it considers high-order information from multiple images

Method	DC [22]	Internet [51]	TDK [6]	GO-FMR [49]	DDCRF [73]	CA [4]	FCA [4]	CSA [4]	DOCS [32]	CoA [19]	AGNN
Car	37.1	64.4	64.9	66.8	72.0	80.0	76.9	79.9	82.7	82.0	**84.0**
Horse	30.1	51.6	33.4	58.1	65.0	67.3	69.1	71.4	64.6	61.0	**72.6**
Airplane	15.3	57.3	46.2	60.4	67.7	72.8	70.6	73.1	70.3	67.0	**76.1**
Avg.	27.5	57.3	46.2	60.4	67.7	70.3	72.8	70.6	73.1	67.7	**77.6**

Table 5: Quantitative results on Internet [51] with mean $\mathcal{J}$ (§4.2.2). We show the per-class performance and an overall average.

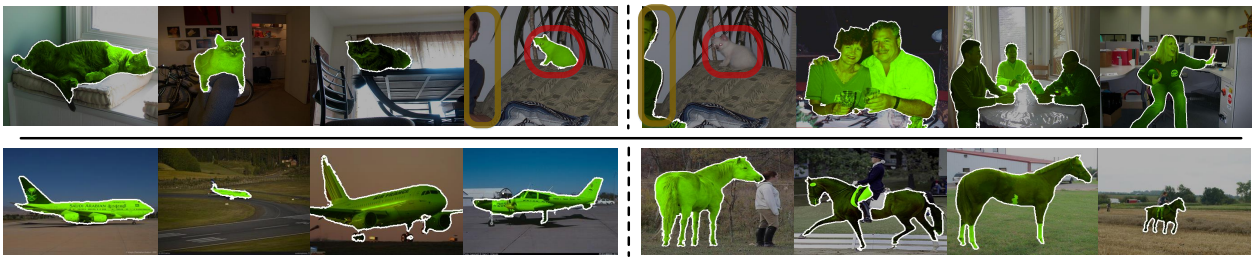

Figure 5: Qualitative image object co-segmentation results on PASCAL VOC [11] (top) and Internet [51] (bottom). See §4.2.3.

Components	Module	DAVIS$_{16}$ mean $\mathcal{J}$	DAVIS$_{16}$ $\Delta\mathcal{J}$
Reference	**Full model** (3 Iterations, N'= 5)	80.7	-
Graph Structure	*w/o.* AGNN	72.2	-8.5
	w/o. Gated Message (Eq. 9)	80.1	0.6
Message Passing	1 iteration	78.7	-2.0
	2 iterations	79.1	-1.6
	4 iterations	80.7	0.0
Input Frames	N'= 3	79.6	-1.1
	N'= 6	80.7	0.0
	N'= 7	80.7	0.0
Post-Process	*w/o.* CRF	78.9	-1.8

Table 6: Ablation study (§4.3) on the val set of DAVIS$_{16}$ [45].

during inference, enabling it to capture richer semantic relations within the image groups.

Internet. We evaluate our model (pre-trained on PASCAL VOC) on Internet [4, 49]. Quantitative results in Table 5 again demonstrate the superiority of AGNN (4.5% performance gain compared with the second best method). The result of AGNN is higher than compared methods for three classes: *Car* (84.0%), *Horse* (72.6%), *Airplane* (76.1%).

4.2.3 Qualitative Results

Fig. 5 shows some sample results. Specifically, the first four images in the top row belong to the *Cat* category (red circle), while the last four images contain the *Person* category (yellow circle) with significant intra-class variation. For both cases, our AGNN successfully detects the common object instances amongst background clutter. For the second row, AGNN also performs well in the cases with remarkable intra-class appearance change.

4.3. Ablation Study

We perform an ablation study on DAVIS$_{16}$ [45] to investigate the effect of each essential component of AGNN.

Effectiveness of Our AGNN. To quantify the contribution of our AGNN, we derive a baseline *w/o. AGNN*, which indicates the results from our backbone model, DeepLabV3. As shown in Table 6, AGNN indeed brings significant performance improvements (72.2→80.6 in term of mean $\mathcal{J}$).

Gated Message Aggregation Strategy. In Eq. 9, we equip the message passing with a channel-wise gated mechanism to decrease the negative influence of irrelevant frames. To evaluate this design, we offer a baseline *w/o. Gated Message*, which aggregates messages directly. A performance degradation is observed after excluding the gates.

Message Passing Iterations K. To investigate the message passing iterations K, we report the performance as a function of Ks. We find that, with more iterations ($1 \rightarrow 3$), better results can be obtained. The performance of the message passing converges at $K = 3$.

Node Numbers N' During Inference. To evaluate the impact of the number of nodes N' during inference, we report performance with different values of N'. We observe that, with more input frames ($3 \rightarrow 5$), the performance raises accordingly. When even more frames are considered ($5 \rightarrow 7$), the final performance does not change obviously. This may be due to the redundant content in video sequences.

5. Conclusion

This paper proposes a novel AGNN based ZVOS framework for capturing the relations among videos frames and inferring the common foreground objects. It leverages an attention mechanism to capture the similarity between nodes and performs recursive message passing to mine the underlying high-order correlations. Meanwhile, we demonstrate the generalizability of AGNN by extending it to IOCS task. Extensive experiments on three ZVOS and two IOCS datasets indicate that our AGNN performs favorably against current state-of-the-art methods. This further illustrates the importance of AGNN which can capture diverse relations among similar video frames or semantically related images.

Acknowledgements This work was supported in part by ARO grant W911NF-18-1-0296, Beijing Natural Science Foundation under Grant 4182056, CCF-Tencent Open Fund, Zhijiang Lab's International Talent Fund for Young Professionals, and the National Science Foundation (CAREER IIS-1253549).

References

[1] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016. 5

[2] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. In *ACL*, 2018. 2

[3] Jiale Cao, Yanwei Pang, and Xuelong Li. Triply supervised decoder networks for joint detection and segmentation. In *CVPR*, 2019. 3

[4] Hong Chen, Yifei Huang, and Hideki Nakayama. Semantic aware attention based deep object co-segmentation. In *ACCV*, 2018. 3, 7, 8

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 4, 5

[6] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, 2014. 8

[7] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017. 1, 2, 3, 6

[8] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015. 6

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014. 5

[10] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*, 2015. 2

[11] Mark. Everingham, Luc. Van Gool, Christopher. K. I. Williams, John. Winn, and Andrew. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2, 7, 8

[12] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 2, 6

[13] Katerina Fragkiadaki, Pablo Arbelaez, Panna Felsen, and Jitendra Malik. Learning to segment moving objects in videos. In *CVPR*, 2015. 2

[14] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, 2017. 2, 3

[15] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *IJCNN*, 2005. 2

[16] Junwei Han, Rong Quan, Dingwen Zhang, and Feiping Nie. Robust object co-segmentation using background prior. *IEEE TIP*, 27(4):1639–1651, 2018. 3

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 6

[18] Dorit. S. Hochbaum and Vikas. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009. 3

[19] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Co-attention cnns for unsupervised object co-segmentation. In *IJCAI*, 2018. 8

[20] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *ECCV*, 2018. 2

[21] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *CVPR*, 2017. 1, 2, 3, 6

[22] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 3, 8

[23] Yeong Jun Koh, Young-Yoon Lee, and Chang-Su Kim. Sequential clique optimization for video object segmentation. In *ECCV*, 2018. 2

[24] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *ICCV*, 2015. 6

[25] Gunhee Kim, P. Eric Xing, Li Fei-Fei, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*, 2011. 3

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 2

[27] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017. 2, 6

[28] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, 2011. 6

[29] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow guided recurrent neural encoder for video salient object detection. In *CVPR*, 2018. 2, 3

[30] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C.-C. Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *CVPR*, 2018. 2, 3

[31] Siyang Li, Bryan Seybold, Alexey Vorobyov, Xuejing Lei, and C.-C. Jay Kuo. Unsupervised video object segmentation with motion-based bilateral networks. In *ECCV*, 2018. 1, 2

[32] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018. 3, 7, 8

[33] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *ICLR*, 2016. 2

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3, 7

[35] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 4

[36] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang. Deep regression tracking with shrinkage loss. In *ECCV*, 2018. 2

[37] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019. 2

[38] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for

video object segmentation. In *ACCV*, 2018. 6

[39] Lopamudra. Mukherjee, Vikas. Singh, and Charles. R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *CVPR*, 2009. 3

[40] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, 2016. 2

[41] Peter Ochs and Thomas Brox. Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011. 1, 2, 6

[42] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE TPAMI*, 36(6):1187–1200, 2014. 2

[43] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 1, 2, 6

[44] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 6

[45] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 2, 6, 8

[46] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 2, 6, 7

[47] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 2, 6

[48] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, 2018. 2

[49] Rong Quan, Junwei Han, Dingwen Zhang, and Feiping Nie. Object co-segmentation via graph optimized-flexible manifold ranking. In *CVPR*, 2016. 3, 7, 8

[50] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs. In *CVPR*, 2006. 3

[51] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013. 2, 3, 7, 8

[52] Jose. C. Rubio, Joan. Serrat, Antonio. Lopez, and Nikos. Paragios. Unsupervised co-segmentation through region matching. In *CVPR*, 2012. 3

[53] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE TNNLS*, 20(1):61–80, 2009. 2, 3

[54] Mennatullah Siam, Chen Jiang, Steven Lu, Laura Petrich, Mahmoud Gamal, Mohamed Elhoseiny, and Martin Jagersand. Video segmentation using teacher-student adaptation in a human robot interaction (hri) setting. In *ICRA*, 2019. 6

[55] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018. 1, 3, 5, 6

[56] Zhiqiang Tao, Hongfu Liu, Huazhu Fu, and Yun Fu. Image cosegmentation via saliency-guided constrained clustering with cosine similarity. In *AAAI*, 2017. 3

[57] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, 2017. 1, 2, 5, 6

[58] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017. 1, 2, 5, 6

[59] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *CVPR*, 2016. 2

[60] Yi-Hsuan Tsai, Guangyu Zhong, and Ming-Hsuan Yang. Semantic co-segmentation in videos. In *ECCV*, 2016. 6

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 4

[62] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 2, 4

[63] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, 2019. 1, 7

[64] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation revisited: Models and optimization. In *ECCV*, 2010. 3

[65] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 3, 7

[66] Wenguan Wang and Jianbing Shen. Higher-order image co-segmentation. *IEEE TMM*, 18(6):1011–1021, 2016. 3

[67] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015. 1, 2

[68] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE TPAMI*, 41(4):985–998, 2018. 2

[69] Wenguan Wang, Hongmei Song, Shuyang Zhao, Jianbing Shen, Sanyuan Zhao, Steven CH Hoi, and Haibin Ling. Learning unsupervised video object segmentation through visual attention. In *CVPR*, 2019. 6

[70] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 4

[71] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 2

[72] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 6

[73] Ze-Huan Yuan, Tong Lu, and Yirui Wu. Deep-dense conditional random fields for object co-segmentation. In *IJCAI*, 2017. 8

[74] Dong Zhang, Omar Javed, and Mubarak Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013. 1, 2

[75] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. 4

[76] Zilong Zheng, Wenguan Wang, Siyuan Qi, and Song-Chun Zhu. Reasoning visual dialogs with structural and partial observations. In *CVPR*, 2019. 2

[77] Wang Ziqin, Xu Jun, Liu Li, Zhu Fan, and Shao Ling. Ranet: Ranking attention network for fast video object segmentation. In *ICCV*, 2019. 3